Amendments to the Specification:

In paragraph [0020]:

Fig.4 shows packet fragmentation of an incoming IP packet 400 into a first outgoing IP packet 402 and a second outgoing IP packet 404 according to a first embodiment of the present invention. An incoming Ethernet frame 406 includes a 14-byte Ethernet header, and a 1500-byte Ethernet payload comprising the incoming IP packet 400. The IP packet 400 includes a 20-byte IP header and a 1480-byte IP payload. As shown in Fig.4, the incoming IP packet 400 is fragmented into the first outgoing IP packet 402 and a second outgoing IP packet 404. The first outgoing IP packet 402 includes a 20-byte first outgoing IP ~~header~~ sub-header and an 8-byte first fragment. The first outgoing IP packet 402 is included in a first outgoing PPPoE frame 408, which additionally includes a 14-byte Ethernet header and an 8-byte PPPoE header. The second outgoing IP packet 404 includes a 20-byte second outgoing IP ~~header~~ sub-header and a 1472-byte second fragment. The second outgoing IP packet 44 is included in a second outgoing PPPoE frame 410, which also includes a 14-byte Ethernet header and an 8-byte PPPoE header. As shown in Fig.4, to make room for the 8-byte PPPoE header, the 1480-byte incoming IP payload is fragmented into the 8-byte first outgoing fragment and the 1472-byte second outgoing IP fragment, which are transmit in that order.

In paragraph [0021]:

Fig.5 shows the usage of a plurality of buffers internal to a gateway, such as the gateway 112 shown in Fig.1, which are used when fragmenting the incoming IP packet according to the present invention. A 14-byte outgoing Ethernet header (Eth) and an 8-byte PPPoE header for the first outgoing PPPoE frame 408 are generated and stored in Buffer 1. The original 20-byte IP header of the incoming IP packet 400 has the MF, Offset, Length, and Checksum fields modified in accordance with the 8-bytes of the first

fragment (Fragment 1), which are stored in the first buffer. After transmitting the first outgoing PPPoE frame 408 (i.e. at the Copy point in Fig.5), a new Ethernet header and a new PPPoE header are generated for the second outgoing PPPoE frame 410. Additionally, the first outgoing IP ~~header~~ sub-header stored in the first buffer needs to have the MF, Offset, Length, and Checksum fields modified in accordance with the remaining 1472-bytes stored in Buffers 1 to 12 as Fragment 2. The new Ethernet header, the new PPPoE header, the second outgoing IP ~~header~~ sub-header, and the first 78-bytes of Fragment 2 are transmit as the beginning of the second outgoing PPPoE frame 410. At this point, the information stored in the first buffer (Buffer 1) is no longer needed and can be freed for use in storing a next incoming packet. Continuing, the next 128-bytes of Fragment 2 stored in the second buffer (Buffer 2) are transmit. As soon as the data in the second buffer (Buffer 2) has been transmitted, Buffer 2 can be freed for use. This process continues for the remaining buffers (Buffer 3 to Buffer 12) with each buffer being freed immediately after having its data transmitted. This is more efficient than the prior art, which requires the information in all the buffers to be stored until the data in the last buffer has been transmitted.

In paragraph [0026]:

Step 606: Modify a header of the incoming packet according to the first fragment to generate a first outgoing ~~header~~ sub-header that corresponds to the first fragment.

In paragraph [0027]:

Step 608: Transmit the first outgoing packet, which is formed by the first outgoing ~~header~~ sub-header and the first fragment.

In paragraph [0028]:

Step 610: Modify either the header of the incoming packet or the first outgoing ~~header~~ sub-header according to the second fragment to generate a second outgoing ~~header~~ sub-header that corresponds to the second fragment.

In paragraph [0029]:

Step 612: Transmit the second outgoing packet, which is formed by the second outgoing ~~header~~ sub-header and the second fragment.

In paragraph [0030]:

It should be noted that the first outgoing packet (step 608) is transmitted before the second outgoing packet (step 612). Additionally, as soon as all the data in the first storage element has been transmitted, the original header of the incoming packet, the first outgoing ~~header~~ sub-header, and the second outgoing ~~header~~ sub-header are no longer required to be stored. For this reason, the first storage unit can be freed for other uses. Likewise, as soon as the data in each of the subsequent storage units has been transmitted, they too can be freed. This is more efficient than the prior art, which requires the information in all the storage units to be preserved until the data in the last buffer has been transmitted.